

Theme I

Software Processes

Focus on Extreme Programming

Learning Goals

- How did agile methods come about?
- What are the principles of agility?
- How are agile processes carried out?
- Can agile processes be combined with non-agile ones?
- In conclusion => understanding of the main ideas of agile development methods and XP in particular.

Table of content

- Definition Software Process
- Waterfall
- Iterative and incremental development
- Agile development
 - principles
 - cycle
 - integration with non-agile processes

Software Process

Rehearsal

- Software project composed of activities or steps
 - Requirements, design, implementation, testing, deployment, maintenance.
- Activities organized into phases
- A software process:
 - prescribes the order and frequency of phases
 - specifies criteria for moving from one phase to the next
 - defines the deliverables of the project
- Consider your last project or programming exercise. How were the activities/steps defined, executed and controlled?

Think/pair/share

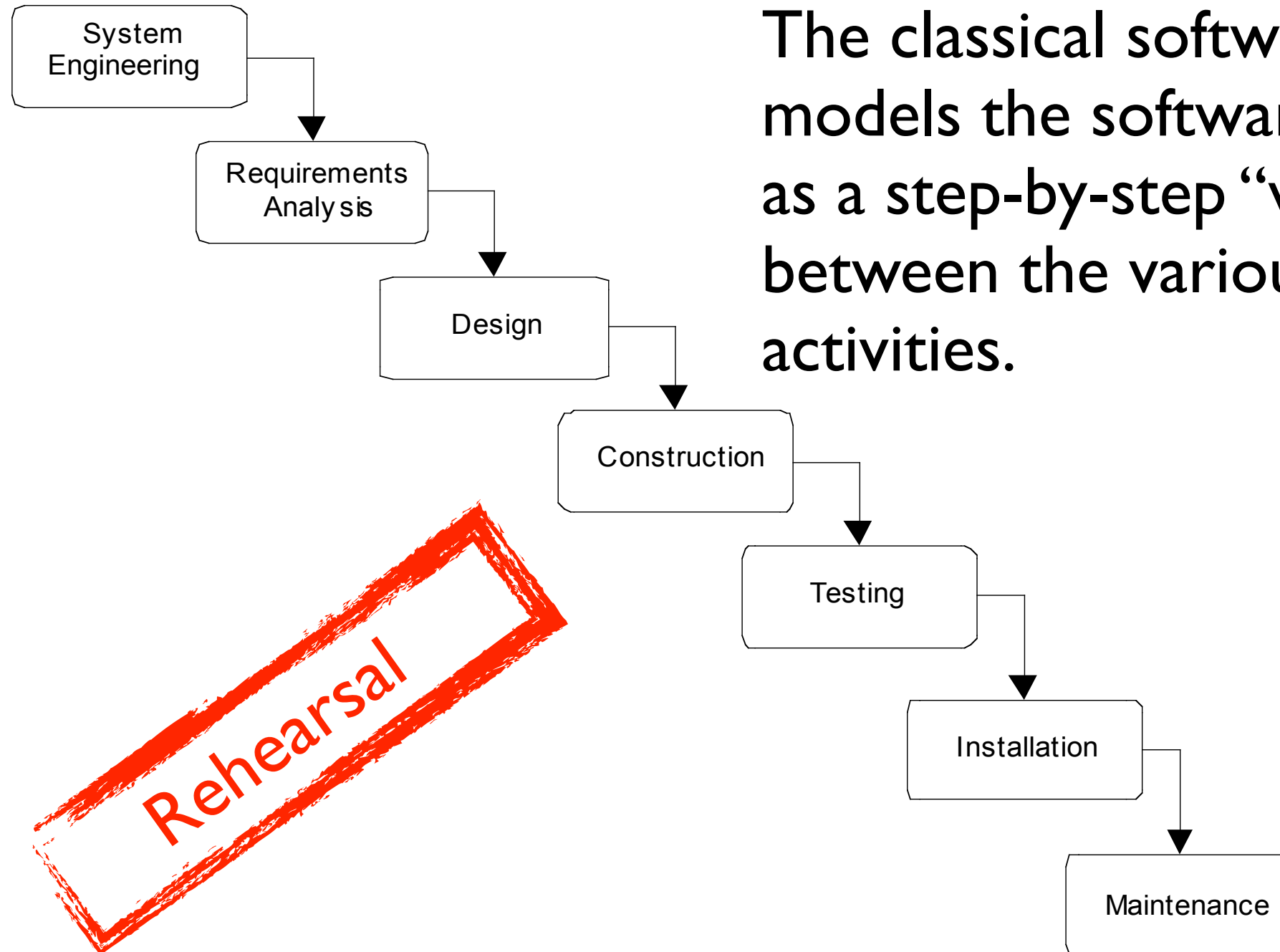


Enumerate the software processes you know, define them using one sentence and indicate the ones you have used in the past.

Énumérez les processus de logiciel que vous connaissez, définissez les processus en une phrase et indiquez les processus que vous avez déjà utilisés.

Waterfall Development

The classical software lifecycle models the software development as a step-by-step “waterfall” between the various development activities.



Iterative and Incremental



- **Iterative**

- repeated execution of the waterfall phases, in whole or in part, resulting in a refinement of the requirements, design and implementation

- **Incremental**

- operational code produced at the end of an iteration
- supports a subset of the final product functionality and features
- Artifacts evolve during each phase
- Artifacts considered complete only when software is released

Agile Development

Manifesto for Agile Software Development

We are uncovering better ways of developing software
by doing and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value
the items on the left more.

The Manifesto for Agile Software Development

Agile Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. exploiter
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

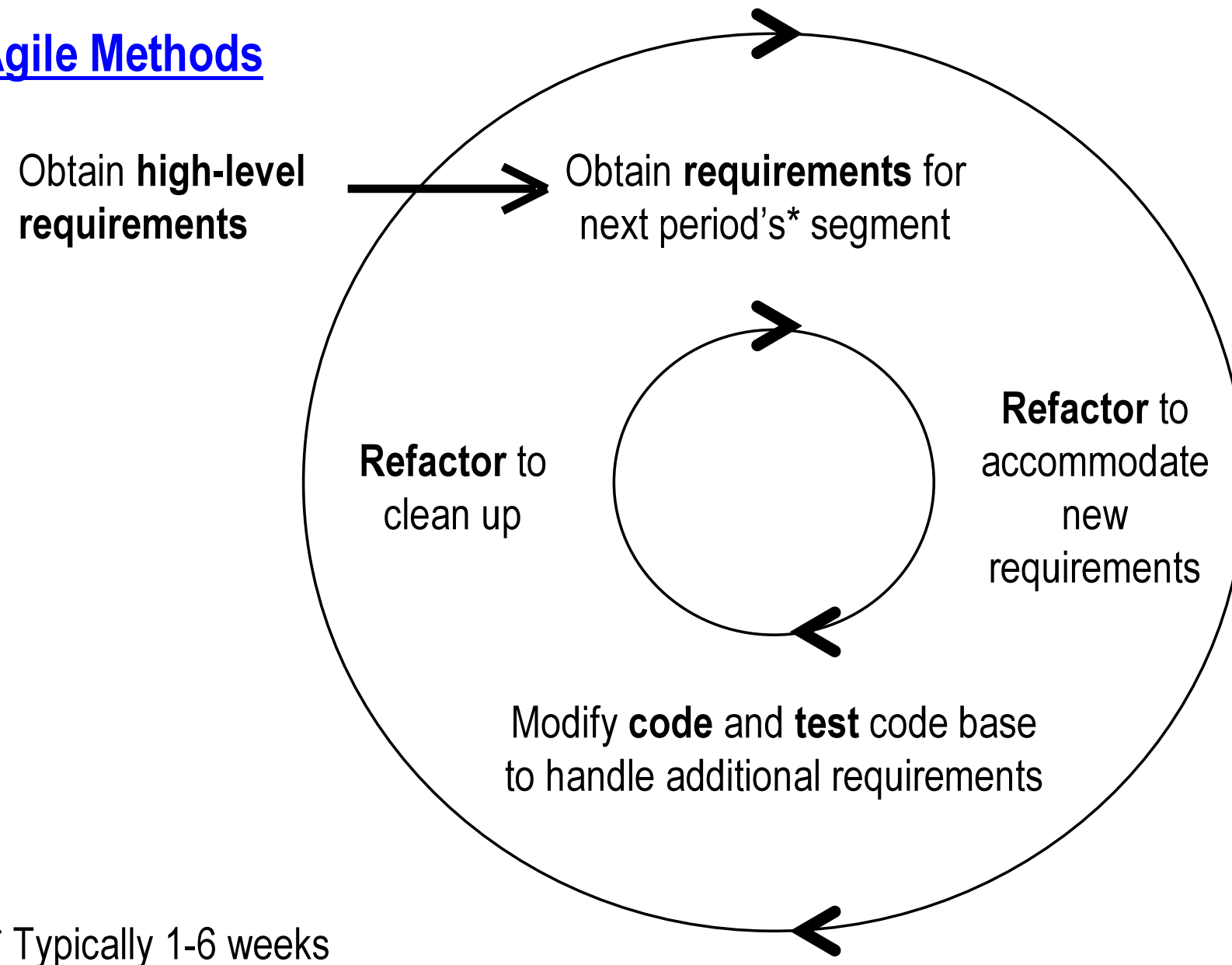
Agile Principles

- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Cycle



Agile Methods

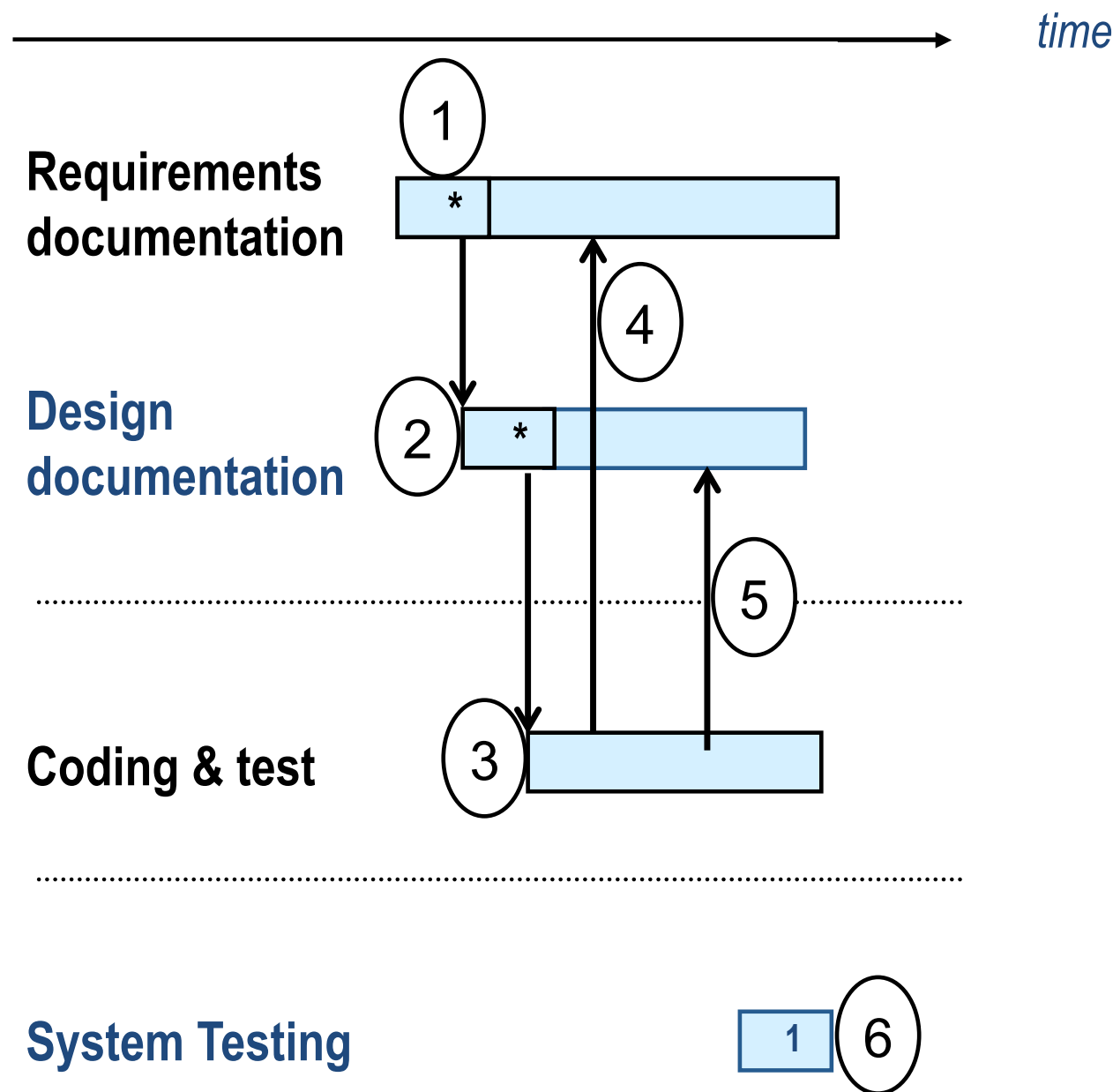


Integrating Agile with Non-Agile Processes



- Regardless of development process used:
 - need to make trade-offs in deciding how extensively to pursue a phase before moving to another phase.
 - E.g. how much effort to spend on planning a software enterprise.

Integrating Agile with Non-Agile Methods: Non-Agile-driven



* High level

Integrating Agile with Non-Agile Methods: Agile-driven



**Initial agile
development**

**Requirements
documentation**

**Design
documentation**

**Coding & test
(including agility?)**

